GPFS Multi-File MPTIO Results

GUPFS Project

Overview

This report documents the timing results of the performance tests done on the GPFS File-system. The primary focus of the performance test was to measure the streaming bandwidth in MB/second

The tests included single-stream and multi-stream tests with different file sizes.

Test Configuration

- We used NERSC MPTIO benchmark to test read and write small (in-cache) and large (out-of-cache) files.
- All tests were run on a quiet system. There were no other activities, neither on the clients nor on the storage controller when the tests were running.

The MPTIO Benchmark

- In MPTIO, a single MPI process is spawned on each node.
- Each node creates M pthreads to perform the actual IO.
- When all threads are ready, the MPI processes perform a barrier sync across the nodes so that all the processes start at about the same time.
  - There is certainly no guarantee this will happen, but generally does when the nodes are not over-subscribed as is the case in our tests.
- The MPI process then waits for all the local threads to complete their IO and synchronizes with another barrier.

Timing in the MPTIO Benchmark

- Each thread records the total amount of elapsed (wall clock) time it took to read/write its entire data region.
- Additionally, each MPI process records the elapsed time from when it started the threads working and when they all completed.
- Finally, the MPI RANK=0 process records the elapsed time between the starting barrier and the ending barrier, signaling that all threads on all nodes have completed.
- The per-thread timings are sent to the MPI RANK=0 process.
- The IO rate is then computed as the total amount of data read/written by all threads divided by the maximum total elapsed time as measured between the MPI barriers by process 0.

Linux Configuration

The four test clients and 4 GPFS NSDs had the following configuration:

- Dual 2.2 GHz Xeon P4 processors, SuperMicro motherboard
- 2 GB 133 MHz ECC memory

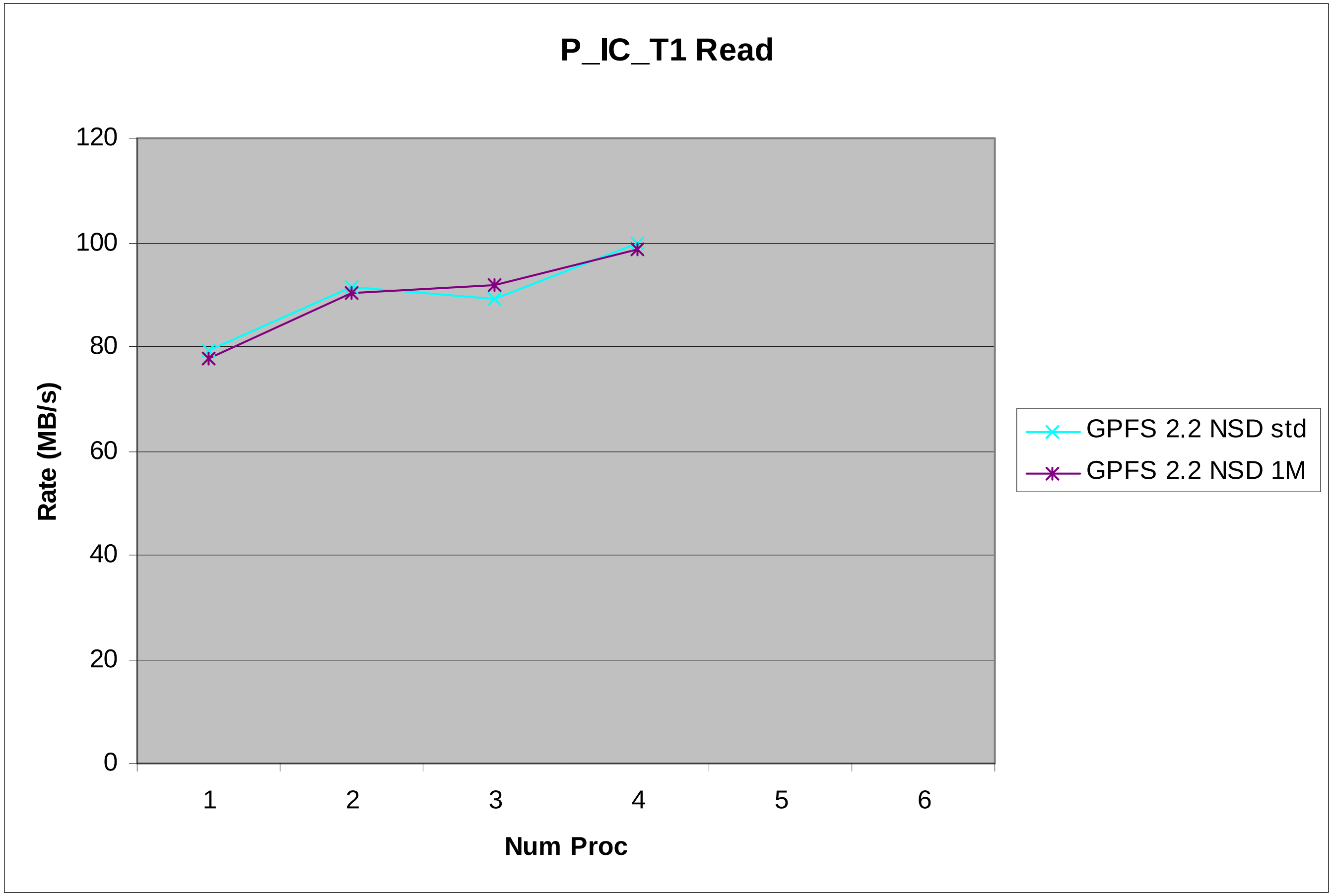
Storage Configuration

- Tests utilized a DDN 8500 storage controller formatted with a 512 byte block size.

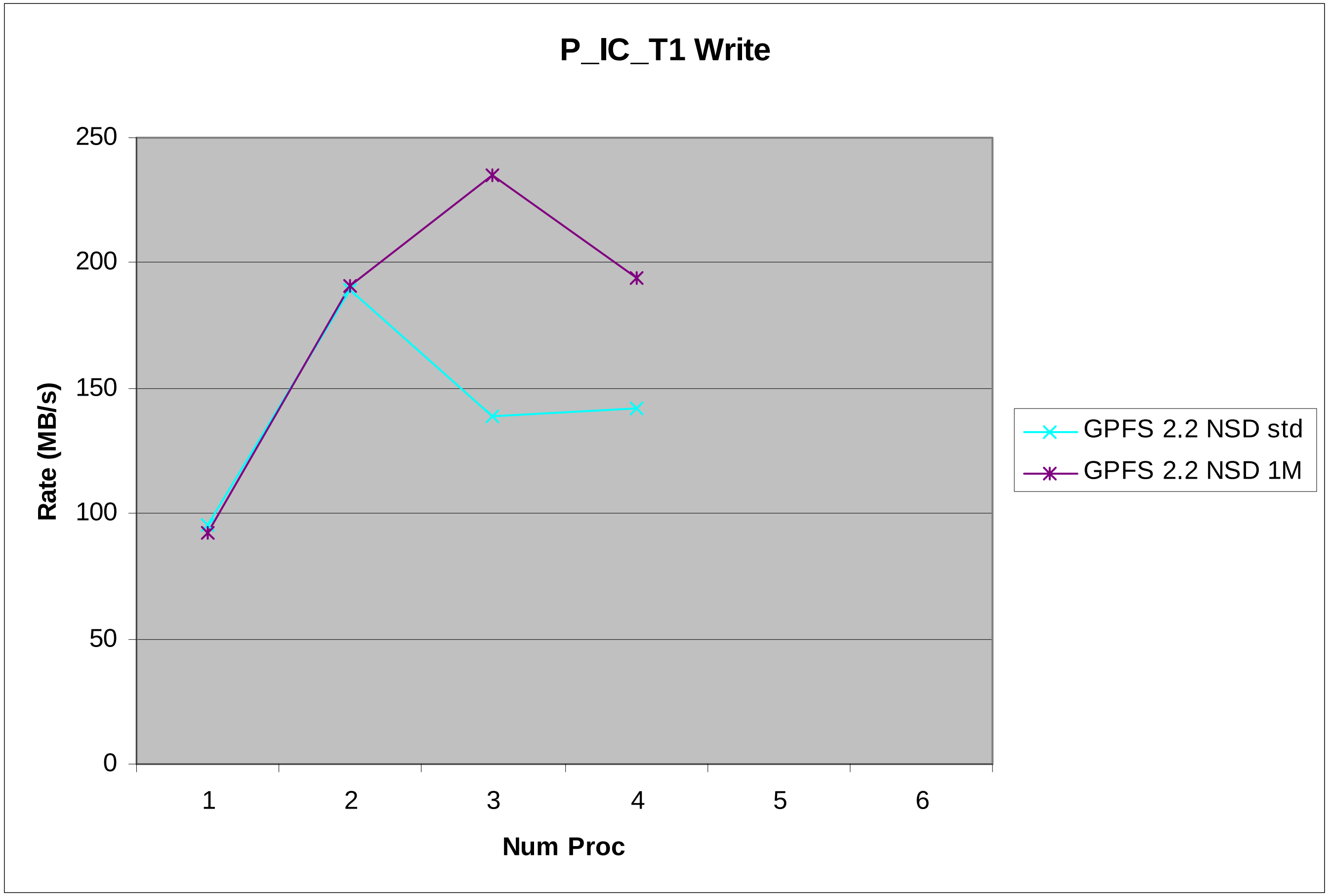
File-system Configuration

- All files were striped across all the NSDs
- Default settings were utilized except where a 1MB file-size is indicated.

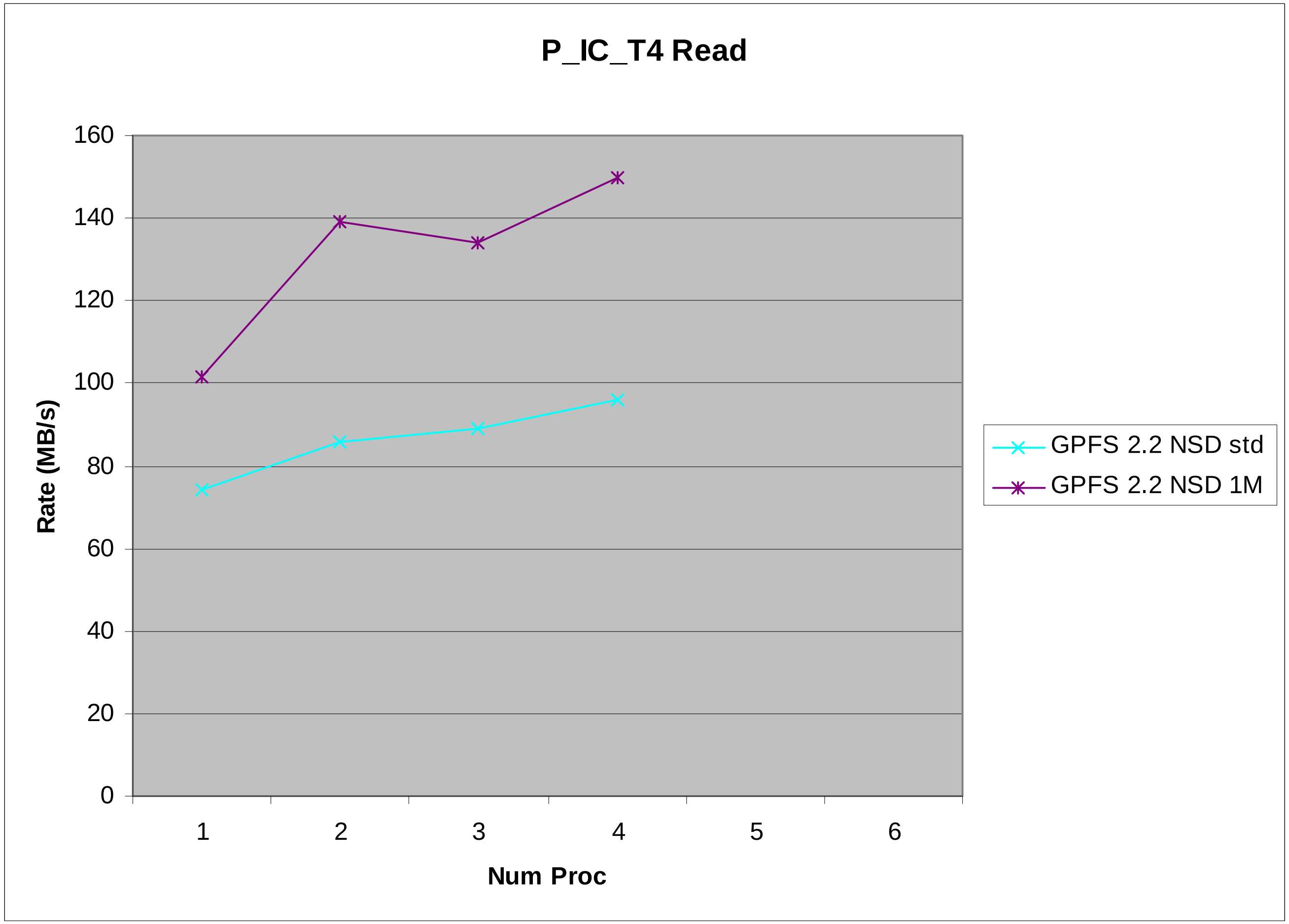
Separate File per Thread, In-cache, Single Thread per Node



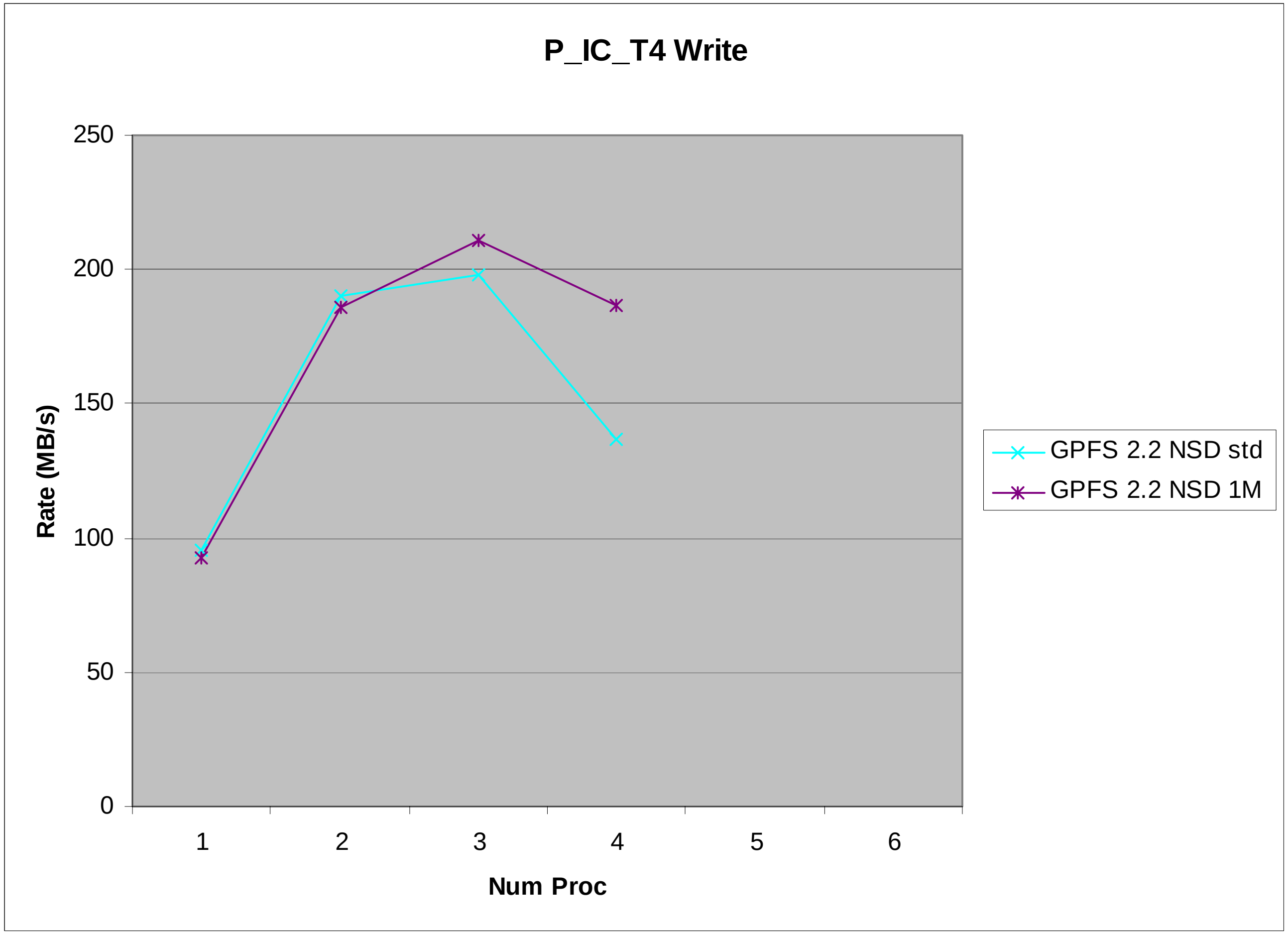
Separate File per Thread, In-cache, Single Thread per Node



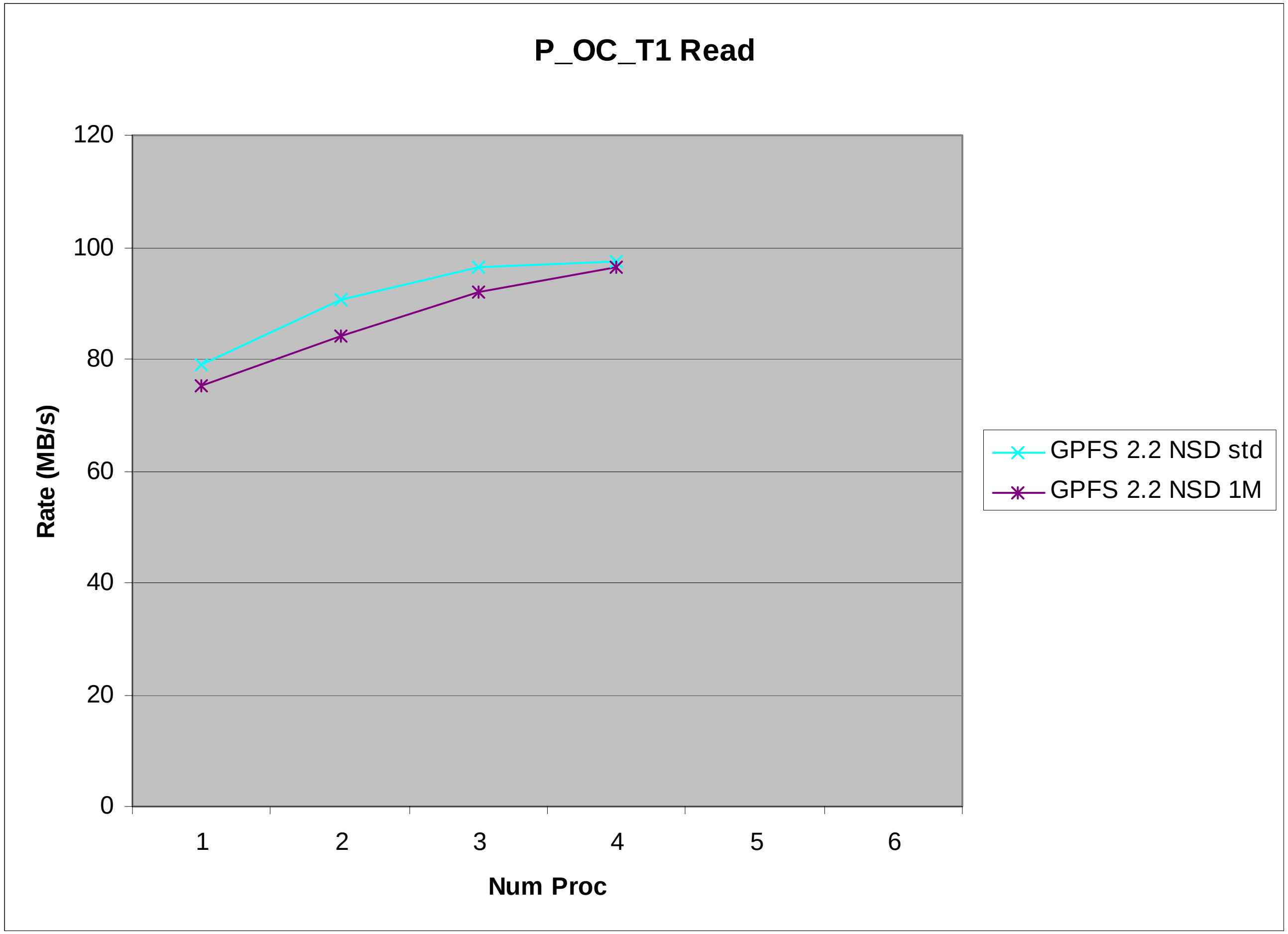
Separate File per Thread, In-cache, Four Threads per Node



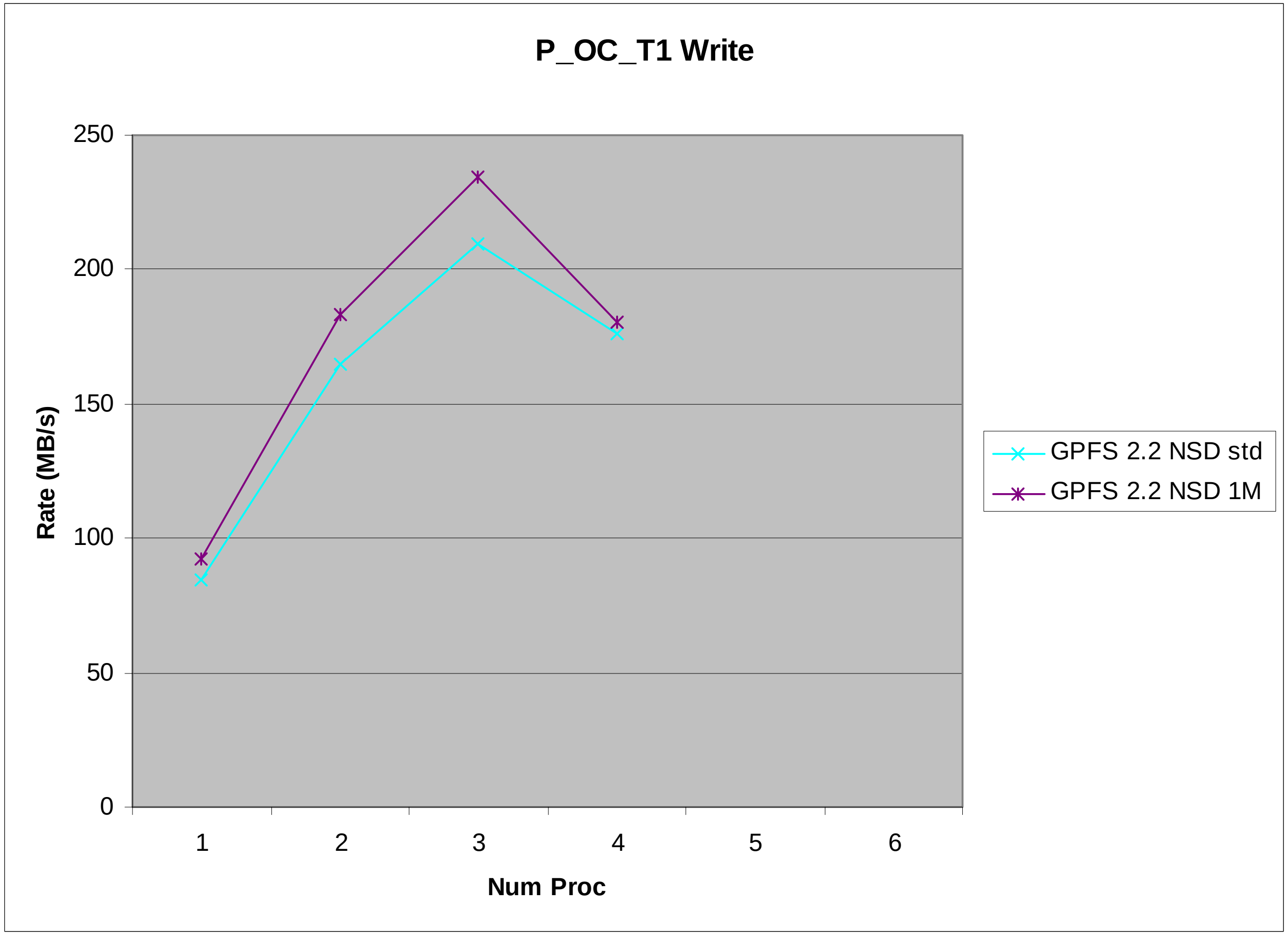
Separate File per Thread, In-cache, Four Threads per Node



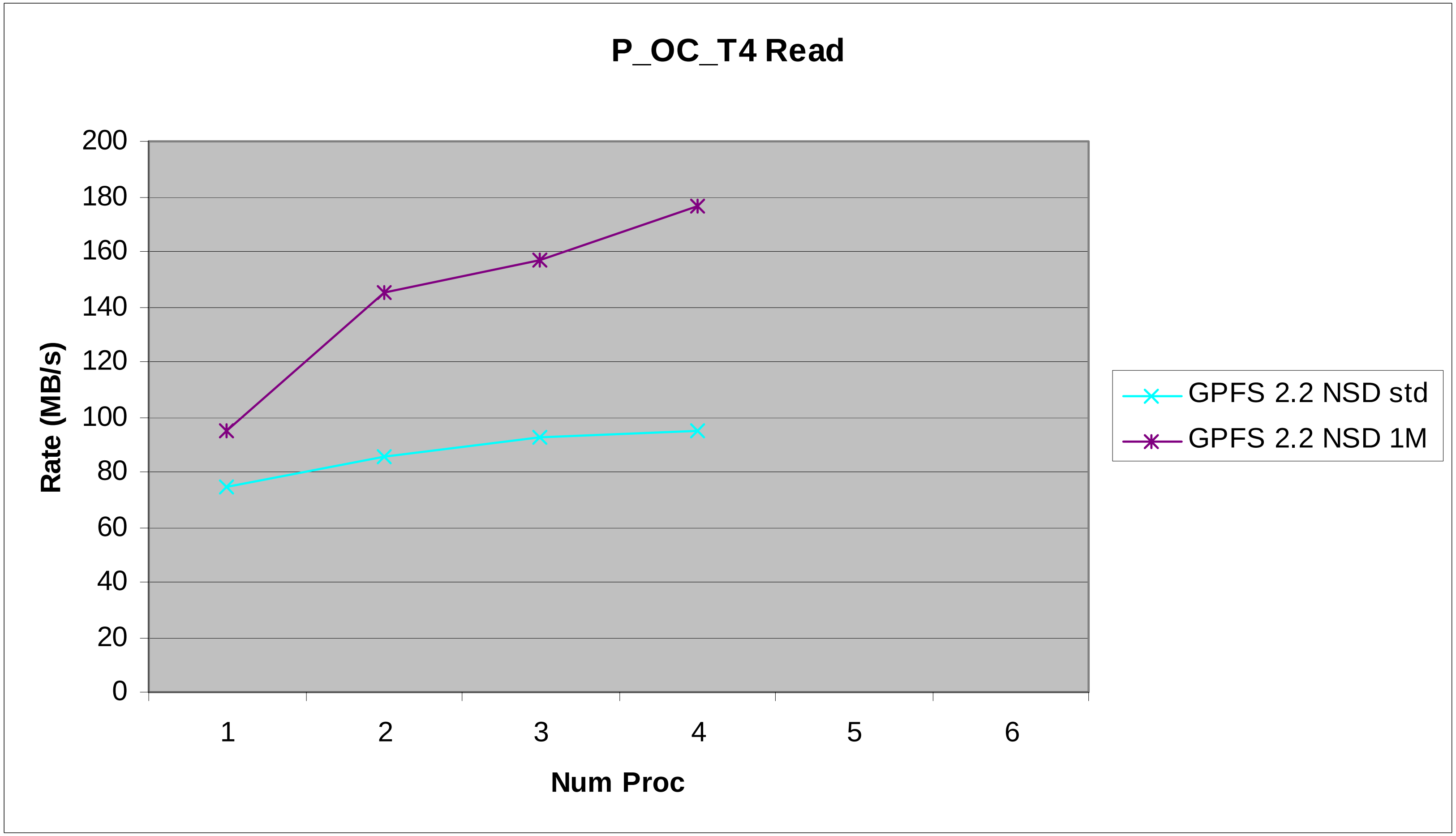
Separate File per Thread, Out-of-Cache, Single Thread per Node



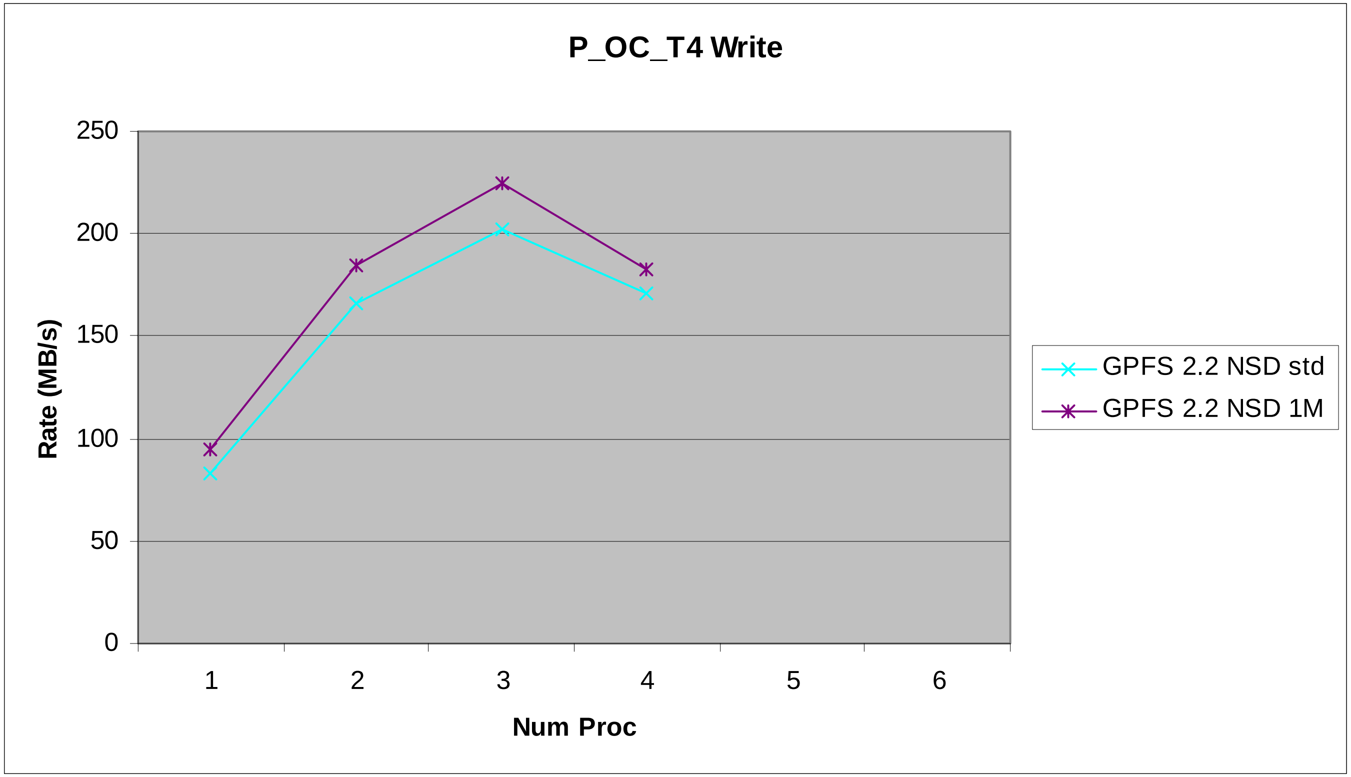
Separate File per Thread, Out-of-Cache, Single Thread per Node



Separate File per Thread, Out-of-Cache, Four Threads per Node



Separate File per Thread, Out-of-Cache, Four Threads per Node



Conclusion

- TBD: Runs in progress